

웹 프로그래밍

8장. 데이터베이스와 JDBC

동아대학교 컴퓨터공학과

■ 오라클 DB 설치

- <https://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html>

Oracle Database Express Edition

스크롤 내려서
Express 에디션
찾아서
다운로드 클릭~

- ☒ Oracle Database 18c Express Edition
- ☒ Oracle Database 11g Release 2 Express Edition for Windows 64
- ☒ Oracle Database 11g Release 2 Express Edition for Linux x86 and Windows

“오라클 사이트 화면은 자주 바뀝니다.
11g 익스프레스 찾으면 됩니다~~”

Oracle Database Express Edition (XE) Release 11.2.0.2.0 (11gR2).

You must accept the [OTN License Agreement for Oracle Database 11gR2 XE](#) to download this software.

☒ **Accept** License Agreement | ☐ **Decline** License Agreement

- Oracle Database 11gR2 Express Edition for Windows x64 [Download](#)
{**Note:** Unzip the download and run the DISK1/setup.exe}
- Oracle Database 11gR2 Express Edition for Windows x32 [Download](#)
{**Note:** Unzip the download and run the DISK1/setup.exe}
- Oracle Database 11gR2 Express Edition for Linux x64 [Download](#)
{**Note:** Unzip the download and the RPM file can be installed as normal}

동의 누른 후
Windows x64
클릭하면
로그인 하라고
나와요.
(아주 간단하게 계정
만들기 가능합니다.)

zip파일
다운로드하고
(300메가 넘음)
압축 풀 후
DISK1폴더에 있는
setup실행

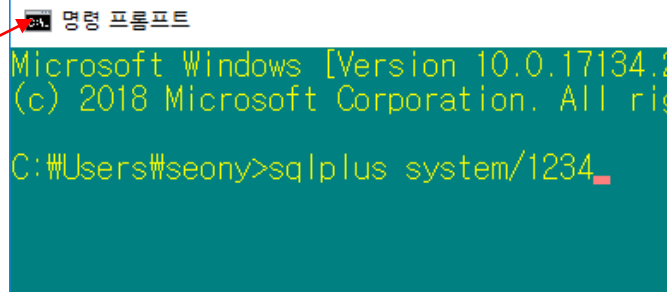
설치 시
비번은 1234

오라클 접속

❖ 설치 끝나면 cmd창 열고 sqlplus system/1234 입력

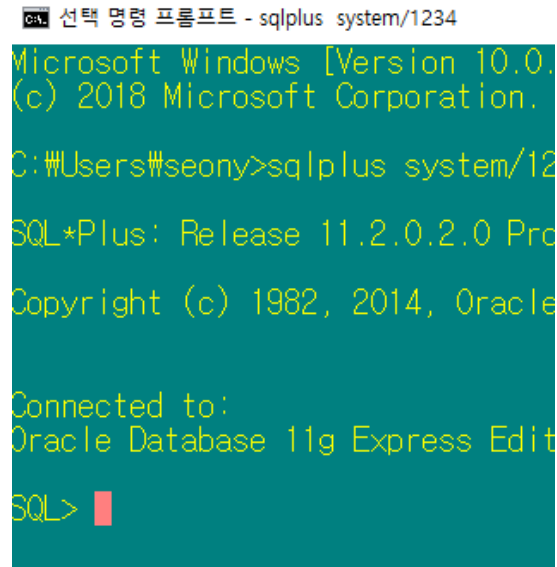
- (system은 관리자 아이디, 1234는 설치 시 입력한 관리자 비번)

"바탕색 및 글자색 바꾸고 싶으면
맨 위 맨 왼쪽 작고 검은 네모
클릭 후 속성 들어가서 바꾸면
됩니다~"



```
명령 프롬프트
Microsoft Windows [Version 10.0.17134.2]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\seony>sqlplus system/1234_
```

❖ 이제 sql 상태가 되었습니다.



```
선택 명령 프롬프트 - sqlplus system/1234
Microsoft Windows [Version 10.0.17134.2]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\seony>sqlplus system/1234
SQL*Plus: Release 11.2.0.2.0 Production
Copyright (c) 1982, 2014, Oracle Corporation
All rights reserved.

Connected to:
Oracle Database 11g Express Edition
SQL>
```

오라클 접속

- ❖ sql 상태에서 일반 사용자 아이디를 만듦 (앞의 system/1234는 관리자임)
 - create user **scott** identified by **tiger**; // 사용자 생성 (scott는 아이디, tiger는 비번)
 - grant connect, resource to **scott**; // 권한 부여
 - grant create view to **scott**; // 권한 부여
 - conn **scott/tiger** // 사용자 연결

```
SQL> conn scott/tiger
Connected.
SQL> _
```

"이제 우리는 scott라는 아이디로 접근하여 SQL을 배워보겠습니다~~"

다음부터는 cmd창에서 sqlplus scott/tiger 하시면 됩니다"

테이블 생성(create) 및 삭제(drop)

```
create table member (  
  name varchar2(10),  
  userid varchar2(10),  
  pwd varchar2(10),  
  email varchar2(20),  
  phone char(13),  
  admin number(1) default 0, -- 0:사용자, 1:관리자  
  primary key(userid)  
);
```

빨간글자는 테이블
파란 글자는 컬럼

혹시 뭔가 잘못되어서 member테이블 삭제하고 싶으면
drop table member; 이렇게 하면 테이블 삭제됨

"sql창에 직접 타이핑하다가 오타나면 ㅋㅋㅋ
메모장에 타이핑 후 복붙하시는 쪽을 추천합니다~~"

레코드 추가(insert) 및 조회(select)

참고: 데이터 한 건을 레코드 1개, 혹은 row 1개라고 부릅니다.

insert into member values

('이소미','somi','1234', 'gmd@naver.com','010-2362-5157',0);

참고: 큰따옴표 말고
작은따옴표 사용하세요

insert into member values

('하상오','sang12','1234', 'ha12@naver.com','010-5629-8888',0);

insert into member values

('김윤승','light','1234', 'yoon1004@naver.com','010-9999-8282',0);

참고: 데이터는 영구 저장되는 것이 아니고
메모리상에만 추가.
영구 저장 안하려면 rollback; 영구 저장하려면 commit;

select name, userid from member; 필요한 컬럼만 보겠다

select * from member; 모든 컬럼 보겠다 (이 때 별표를 all 이라고 부릅니다. "선택트 올 ~")

select * from member where pwd='2222'; 특정 레코드만 보겠다

레코드 수정(update) 및 삭제(delete) p.416

```
update member set phone='010-111-2222'  
where userid='somi';
```

somi 전화번호 변경.

where 부분 없으면 모든 데이터의 전화번호가 일괄수정됨

```
delete member  
where userid='somi';
```

somi 데이터 삭제.

where 부분 없으면 모든 데이터 삭제

"delete 해도 되고 delete from 해도 됩니다."

처음에는

create table 과 insert,

drop table 과 delete 가 헷갈릴 수 있어요~~

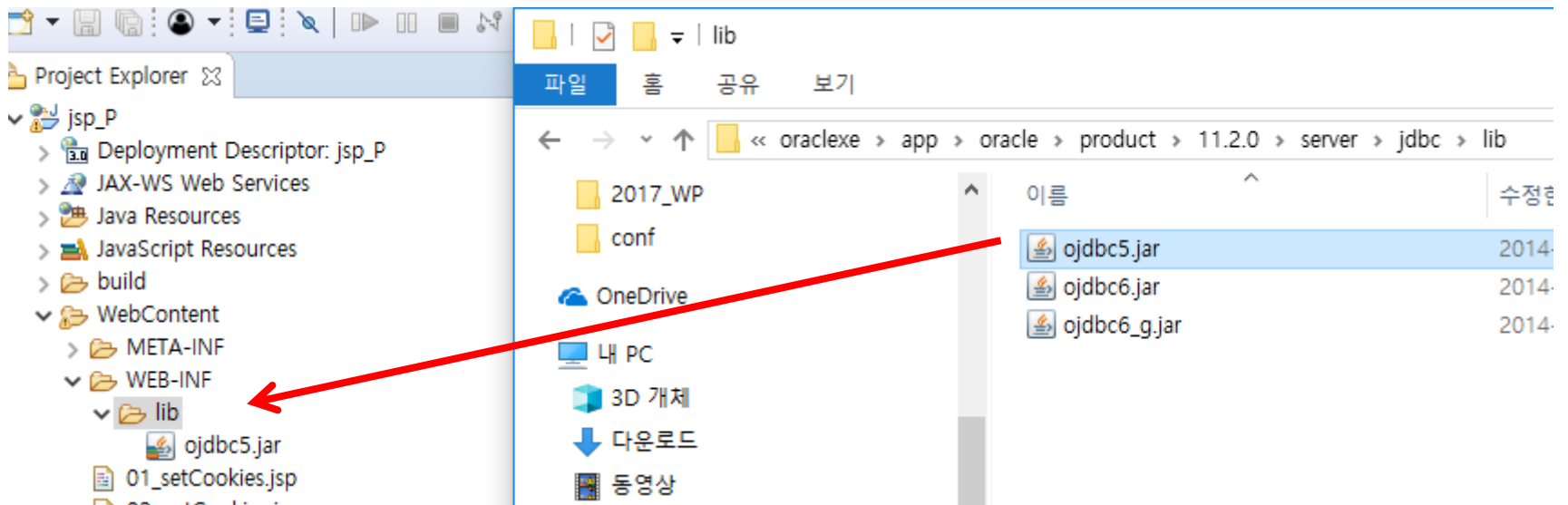
❖ JDBC : Java Database Connectivity

api

- 데이터베이스를 다루기 위한 자바 API
- JDBC가 있음으로 해서 DBMS 상관없이 SQL문을 던져서 실행
 - Oracle을 설치하든 MySQL을 설치하든 상관없이 일관성 있게 사용 가능
- Driver 로드 → Database 연결 → SQL문 실행 순서

JDBC 드라이버 연결 p.420

C:\Woraclexe\Wapp\Woracle\Wproduct\W11.2.0\server\Wjdbc\Wlib
폴더에 있는 ojdbc5.jar파일을 끌어다가 이클립스 lib 에 넣음



핵심 코딩 부분

❖ 아래 5가지가 잘 있는지 확인

- ① `Class.forName` → 드라이버 로드
- ② `DriverManager.getConnection(url, uid, pass)` → DB 연결
- ③ `createStatement()` 나 `prepareStatement()` 둘 중 선택 → SQL문 답을
준비 3c 3p
- ④ `executeQuery(sql)` (select 경우. ResultSet으로 받아야) 혹은
`executeUpdate(sql)` (i/u/d 경우) → SQL문 수행 4Q 4U
- ⑤ `next()` → select 경우 조회된 데이터들 중 한 row씩 정보 가져옴

select는 5번까지
i/u/d는 4번까지 확실히 있어야 함

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
```

```
<%@ page import="java.sql.*"%>
```

```
<%
```

```
String url = "jdbc:oracle:thin:@localhost:1521:XE";
```

```
String uid = "scott"; String pass = "tiger";
```

```
String sql = "select * from member";
```

```
try {
```

```
① Class.forName("oracle.jdbc.driver.OracleDriver");
```

```
② Connection conn = DriverManager.getConnection(url, uid, pass);
```

```
③ Statement stmt = conn.createStatement(); // 3c로 했음. 3p는 나중에
```

```
④ ResultSet rs = stmt.executeQuery(sql); // select니까 executeQuery (4Q)
```

```
%>
```

```
<table width="800" border="1">
```

```
<tr><th>이름</th><th>아이디</th><th>암호</th><th>이메일</th><th>전화번호</th><th>구분</th></tr>
```

```
⑤ <%while (rs.next()) {
```

```
out.println("<tr>");
```

```
out.println("<td>" + rs.getString("name") + "</td>");
```

```
out.println("<td>" + rs.getString("userid") + "</td>");
```

```
out.println("<td>" + rs.getString("pwd") + "</td>");
```

```
out.println("<td>" + rs.getString("email") + "</td>");
```

```
out.println("<td>" + rs.getString("phone") + "</td>");
```

```
out.println("<td>" + rs.getInt("admin") + "</td>"); // getString 해도 됨
```

```
out.println("</tr>");
```

```
}//while의 끝
```

```
%>
```

```
</table>
```

```
<%} catch (Exception e) {
```

```
out.print("죄송합니다. 시스템상 문제가 생겼어요 <br>" + e.getMessage());
```

```
}
```

```
%>
```

p.419

member테이블의
모든 데이터를
웹화면에 출력

컬럼명 대신 컬럼 순서(인덱스)로 해도 됨
예를 들어 rs.getString("name") 대신에
rs.getString(1) 해도 됨

(1) 조회가 잘 된 경우

이름	아이디	암호	이메일	전화번호	구분
이소미	somi	1234	gmd@naver.com	010-2362-5157	0
하상오	sang12	1234	ha12@naver.com	010-5629-8888	0
김윤승	light	1234	yoony1004@naver.com	010-9999-8282	0

(2) 예외처리 발생 경우: 예를 들어 테이블 이름 잘못 입력 시

죄송합니다. 시스템상 문제가 생겼어요
ORA-00942: table or view does not exist

note: 앞에서 commit/rollback에 대해서 언급했었는데요,

참고: 데이터는 영구 저장되는 것이 아니고
메모리상에만 추가.
영구 저장 안하려면 rollback; 영구 저장하려면 commit;

혹시 sql창에서 insert후 jsp돌려서 select 하면
방금 insert한 내용이 안 보일 수도 있습니다.

방금 inser한 내용이 아직 커밋되지 않았기 때문입니다.
(한글문서 작성 후 저장버튼 누르기 전과 비슷)

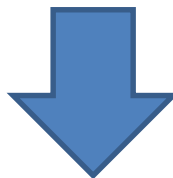
sql창에서 commit; 눌러 주신 후에 jsp 수행 결과 다시
보시면, insert한 자료가 보일 것입니다.

sql창에서만 작업하거나, 이클립스로만 작업할 때는
커밋 신경 안써도 되지만,
섞어서 사용할 때는
sql창에서 데이터 i/u/d 후에 commit; 눌러주세요~

4Q 내부 설명

`ResultSet rs = stmt.executeQuery(sql);`

	name	userid	pwd	email	phone	admin
Begin Of File: Before the First Row						
	이소미	somi	1234	gmd@naver.com	010-2362-5157	0
	하상오	sang12	1234	ha12@naver.com	010-5629-8888	1
	김윤승	light	1234	youn1004@naver.com	010-9999-8282	0
End Of File(EOF) After the Last Row						



Begin Of File((BOF) : Before the First Row					
이소미	somi	1234	gmd@naver.com	010-2362-5157	0
하상오	sang12	1234	ha12@naver.com	010-5629-8888	1
김윤승	light	1234	youn1004@naver.com	010-9999-8282	0
End Of File(EOF) : After the Last Row					

코드 셋 객체가
언어지자 마자 Cursor
위치

←

rs.next()

rs.next() 한 번에 row 한 개.

그 row 안에서 각 칼럼 가져올 때 getString() 혹은 getInt()

name	userid	pwd	email	phone	admin
이소미	somi	1234	gmd@naver.com	010-2362-5157	0
하상오	sang12	1234	ha12@naver.com	010-5629-8888	1
김윤승	light	1234	youn1004@naver.com	010-9999-8282	0

이번에는 insert해볼까요.

“select와 비교해서
뭐가 달라졌나요?”

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ page import="java.sql.*"%>
<%
String url = "jdbc:oracle:thin:@localhost:1521:XE";
String uid = "scott"; String pass = "tiger";
String sql =
    "insert into member values('냥','nyang','9999', 'nyang@donga.ac.kr','010-12-34',0)";
try {
① Class.forName("oracle.jdbc.driver.OracleDriver");
② Connection conn = DriverManager.getConnection(url, uid, pass);
③ Statement stmt = conn.createStatement(); // (3c)
④ stmt.executeUpdate(sql); // insert니까 executeUpdate(4U)
    out.print("<h3>회원 가입 성공 </h3>");
    out.print("<a href='select.jsp'>회원 전체 목록 보기</a>");
} catch (Exception e) {
    out.print("죄송합니다. 시스템상 문제가 생겼어요 <br>" + e.getMessage());
}
%>
```

userid가 기본키인 거 기억하시죠? 같은 userid로 넣으려고 시도하면
아래처럼 예외처리 발생!!

```
죄송합니다. 시스템상 문제가 생겼어요
ORA-00001: unique constraint (SCOTT.SYS_C006997) violated
```

note: update, delete는 insert 기준으로 하시면 됩니다.

select는 여러 건 select할 수도 있고, 1건만 select할 수도 있습니다.

여러 건 경우, 앞의 예처럼 반복문으로 rs.next() 가져오며, 1건 select 경우 rs.next()를 한 번만 하면 됩니다.

앞으로 여러 시스템을 만들어보면서 차차 익숙해져 가도록 하겠습니다.